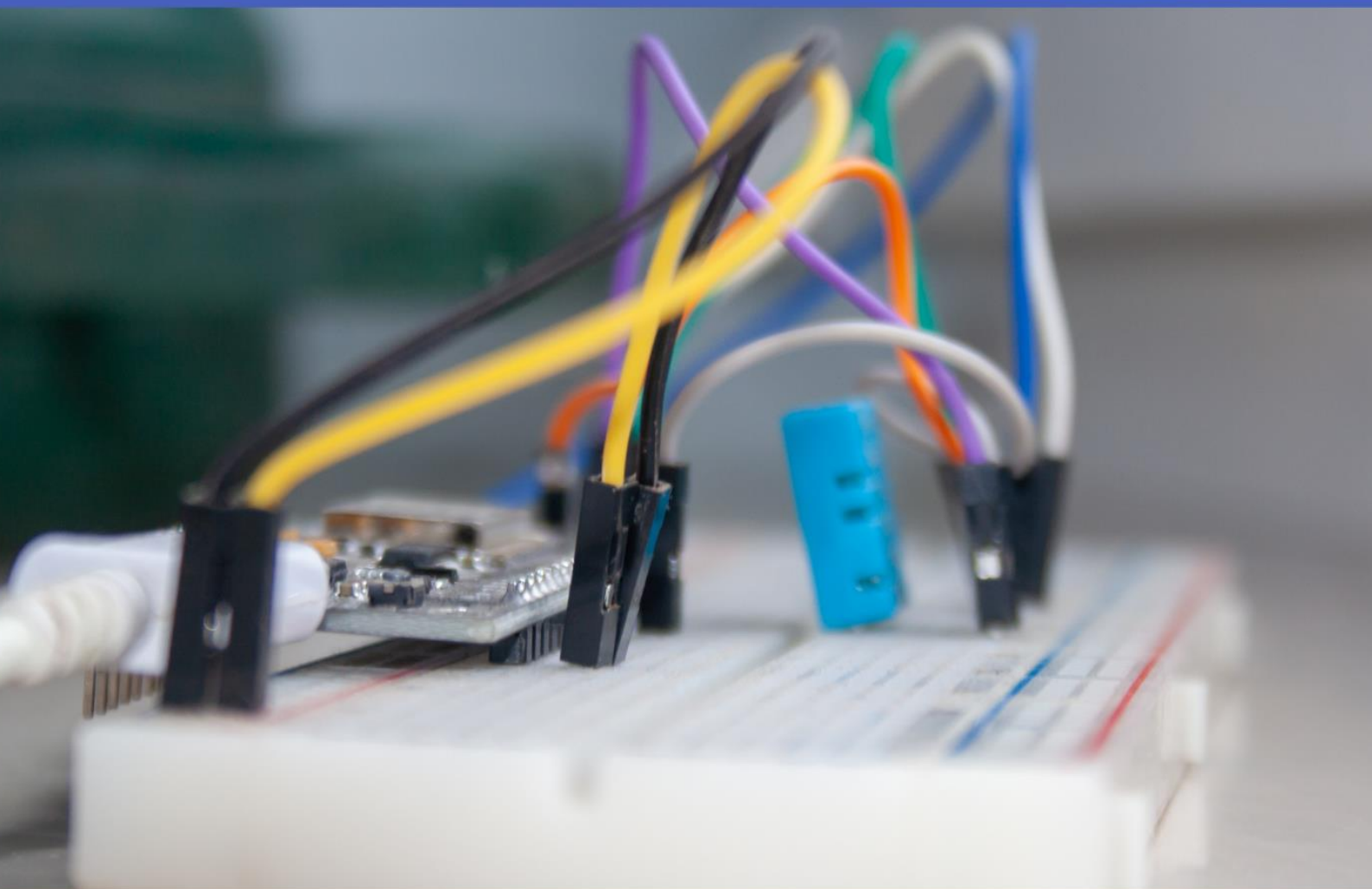


PRAKTYCZNE ASPEKTY NAUCZANIA PROGRAMOWANIA Z WYKORZYSTANIEM MIKROKONTROLERÓW NA LEKCJACH INFORMATYKI W KLASACH IV-VIII



Wzorcowy materiał szkoleniowy w zakresie innowacyjnych rozwiązań organizacyjno-dydaktycznych dla nauczycieli i studentów studiów pedagogicznych

Informatyka w klasach IV-VIII szkoły podstawowej

Łukasz Dworski

Autor:

Łukasz Dworski

Wydawca:

Euro Innowacje sp. z o.o.

Publikacja została opracowana w ramach projektu pt. „Szkoła Ćwiczeń w Gminie Rawicz”, realizowanego w partnerstwie przez Gminę Rawicz (Beneficjent projektu) oraz Euro Innowacje sp. z o.o. (Partner projektu).

Projekt jest finansowany ze środków budżetu państwa oraz Unii Europejskiej, w ramach Europejskiego Funduszu Społecznego, Programu Operacyjnego Wiedza Edukacja Rozwój (PO WER), II Osi Priorytetowej „*Efektywne polityki publiczne dla rynku pracy, gospodarki i edukacji*”, Działania 2.10 „*Wysokiej jakości system oświaty*”.

Publikacja jest udostępniona na zasadach wolnej licencji Creative Commons – Uznanie autorstwa 3.0 Polska (CC BY 3.0 PL).

SPIS TREŚCI

WSTĘP	4
CEL PUBLIKACJI	7
1. PROGRAMOWANIE – EDUKACYJNA INNOWACJA PRZYSZŁOŚCI	10
1.1 <i>Rozwijanie myślenie komputacyjnego</i>	12
1.2 <i>TIK w nauczaniu informatyki.....</i>	14
1.3 <i>Metoda projektu jako strategia uczenia się</i>	16
2. PROGRAMOWANIE MIKROKONTROLERÓW	19
2.1 <i>Charakterystyka platformy Micro:bit</i>	20
2.2 <i>Programowanie z platformą Arduino</i>	21
2.2.1 <i>Zalety i możliwości wykorzystania Arduino</i>	22
3 SYMULATOR ŚWIATEŁ ULICZNYCH	23
3.1 <i>Wykaz elementów układu symulatora</i>	26
3.2 <i>Podłączanie i programowanie modułu Micro:bit</i>	27
3.3 <i>Podłączanie i programowanie obwodu z Arduino</i>	30
PODSUMOWANIE	35
BIBLIOGRAFIA Z UWZGLĘDNIENIEM NETOGRAFII.....	37
WYKAZ TABEL	39
WYKAZ ILUSTRACJI.....	39



WSTĘP

*Sama technologia się nie liczy.
To, co jest ważne, to wiara w to,
że ludzie są dobrzy i mądrzy
i że jeśli damy im nowoczesne narzędzia,
będą w stanie z nimi czynić prawdziwe cuda.*

Steve Jobs

Niniejsza publikacja powstała z myślą o nauczycielach szkół podstawowych II etapu edukacyjnego. Adresatem materiału szkoleniowego są nauczyciele informatyki, nauczyciele, których celem nie jest pogłębianie znajomości wybranej dziedziny zastosowań komputerów i informatyki, a poszerzenie horyzontów poprzez ukazanie młodemu człowiekowi możliwie pełnego obszaru tej dyscypliny.

Efektom zajęć z informatyki w szkole powinno być rozbudzenie wśród uczniów zainteresowania samą informatyką, a nie komputerami.

Nauczyciel, diagnozując umiejętności uczniów, ich możliwości i potrzeby, może tak organizować ich pracę, aby uczniowie osiągnęli cele wykraczające poza treści określone w podstawie programowej.

Główne cele nauczania podczas zajęć z informatyki w szkole podstawowej w klasach 4–8 określone w obowiązującej podstawie programowej to:

- kształcenie umiejętności rozwiązywania problemów z różnych dziedzin z wykorzystaniem metod i narzędzi wywodzących się z informatyki oraz na lepsze zrozumienie, jakie są obecne możliwości technologii, komputerów,
- kształcenie umiejętności programowania. Z tą nauką wiąże się nie tylko czysto techniczna umiejętność, ale przede wszystkim kształcenie logicznego myślenia, precyzyjnego prezentowania pomysłów, organizacji pracy i współpracy, informatyczne podejście do rozwiązywania problemów,
- umożliwienie rozwoju uczniów w przejściu z pozycji „cyfrowych konsumentów” na pozycję „cyfrowych twórców”.



Od klasy 4 zajęcia informatyki zaczynają mieć charakter bardziej formalny. Uczniowie nadal zajmują się różnymi sytuacjami problemowymi, przedstawianymi w sposób opisowy, w tym za pomocą ilustracji i historyjek, ale tworzą je samodzielnie i abstrahują z nich działania, które składają się na własne realizacje w postaci programów lub czynności wykonywanych w innych programach. Rozwijają w ten sposób podejście algorytmiczne przy rozwiązywaniu różnorodnych sytuacji problemowych z różnych dziedzin. Posługują się komputerem, rozwijając również umiejętności wyrażania swoich myśli i ich prezentacji, które wykonują indywidualnie, a także zespołowo, w tym przy realizacji projektów dotyczących problemów z różnych dziedzin. W sieci poszukują informacji przydatnych w rozwiązywaniu stawianych zadań i problemów.

Doceniają rolę współpracy w rozwoju swojej wiedzy i umiejętności. Postępują odpowiedzialnie i etycznie w środowisku komputerowo-sieciowym.

Od klasy 7 dotychczas zdobyte wiedza i umiejętności informatyczne są rozwijane i poszerzane oraz stawiane są pierwsze kroki w tekstowym języku programowania.

Przy użyciu dostępnego oprogramowania uczniowie realizują projekty i rozwijają kompetencje zespołowego rozwiązywania problemów pochodzących z różnych dziedzin. W trakcie prac nad projektami (indywidualnymi lub zespołowymi) uczniowie powinni mieć również możliwość korzystania z komputerów lub innych urządzeń cyfrowych, w zależności od potrzeb wynikających z charakteru zajęć, realizowanych celów i tematów.

Musimy zdawać sobie sprawę z tego, że współpraca, niełatwa sztuka prowadzenia dyskusji, przedstawiania argumentów, nazywanie i wyrażanie emocji, także tych trudnych... to umiejętności, nad których rozwojem zawsze warto pracować. Po okresie edukacji zdalnej powinniśmy tę strefę rozwoju dziecka wspierać szczególnie. Zaprezentowane scenariusze łączą rozwój umiejętności społecznych z logicznym, algorytmicznym myśleniem, zadaniowym podejściem do problemów oraz stopniowym wprowadzaniem podstawowych



pojęć, metod programistycznych. Propozycje zajęć zaproponowane w publikacji bazują na wykorzystywaniu różnorodnych narzędzi, zarówno tych offline, jak i tych cyfrowych.



CEL PUBLIKACJI

Celem głównym opracowania jest przedstawianie różnych aspektów rozwijania i wspierania kompetencji kluczowych u uczniów ze szczególnym uwzględnieniem innowacyjnych metod do realizacji zadań w interdyscyplinarnych projektach informatycznych i technicznych dla uczniów klas IV-VIII szkół podstawowych.

Nauka programowania jako element budowania kompetencji cyfrowych jest niezbędna albowiem:

- programowanie rozwija kreatywność i umiejętność logicznego myślenia,
- programowanie powinno być adresowane do wszystkich uczniów, przez wszystkie lata w szkole,
- powinno być umieszczone w szerszym kontekście kształtowania pojęć i poznawania informatycznych metod rozwiązywania problemów za pomocą komputerów,
- środowisko programowania to współczesne środowisko aplikacji komputerowych,
- umiejętność programowania jest przydatna w innych dziedzinach,
- zalecana metodyka obejmuje w swoim zakresie następujące kroki postępowania: sytuacja problemowa – abstrakcja i pojęcia – algorytm – metoda rozwiązywania – program.

Jarosław Drapała¹ pisze o nauce programowania jako elemencie kształcenia podstawowego. Uważa on, że nauczanie programowania należy traktować jako umiejętności zawodowe, a także nauczanie programowania jako umiejętności podstawowej obok pisania, czytania i liczenia oraz nauczanie programowania jako droga do rozwijania zdolności intelektualnych. Jest także propagatorem oryginalnej idei: „Nauczanie programowania służy aktywnemu zdobywaniu

¹ <http://www.pryzmat.pwr.edu.pl/sitepages/ludziePolitechniki.aspx?i=51> [Dostęp: 03.03.2022]



wiedzy o mechanizmach funkcjonowania świata”², czyli podkreślenie roli prostoty języka, z której wyłania się program wykonujący skomplikowane działania. Możliwe i celowe jest sprzęgnięcie nauki programowania z innymi działami wiedzy, a programowanie symulacji komputerowych daje możliwość samodzielnej weryfikacji nabywanej wiedzy, w tym świadomość założeń stojących u jej podstaw. Umiejętność programowania Deptuła uważa za narzędzie do samodzielnego wytwarzania nowej wiedzy, której dobrym przykładem jest realizacja idei w ramach projektu „Język Maszyn”, której jest współautorem. Dlatego należy przytoczyć słowa Alvina Tofflera, który pisze: „Analfabetami XXI wieku będą ci, którzy nie umieją się uczyć, oraz ci, którzy nie są w stanie oduczyć się tego, czego już się nauczyli i uczyć się na nowo”³. Czy nauka programowania jest nowym wyzwaniem, zadaniem edukacji? Czy mamy do czynienia z kompleksowym kształceniem informatycznym w formie kompetencji cyfrowych poprzez obowiązkową naukę programowania w szkole? W raporcie autorstwa Anny Mazgal i Alka Tarkowskiego już na wstępie czytamy: Programowanie z pewnością wpisuje się najszerzej w model umiejętności praktycznych, jednak rosnąca rola technologii sprawia, że i użytkownicy w coraz większej mierze będą musieli zdobyć tego rodzaju kompetencje, zwłaszcza w zawodach kreatywnych oraz w innowacyjnych przedsięwzięciach, gdzie dostępne oprogramowanie może nie tylko być dopasowywane do nowych lub specyficznych potrzeb. Autorzy postrzegają naukę programowania jako wyzwanie w kontekście nierówności społecznej, będącej, zdaniem autorów, barierą w dostępie do edukacji⁴.

² Kujawsko-Pomorski Przegląd OświatowyUczMy nr 2/2020 https://www.kpcen-torun.edu.pl/files/60b91ae29eab12-34_marzec-kwiecien_2020.pdf [Dostęp: 04.03.2022]

³ <https://www.robotlab.com/blog/a-story-around-this-pic-> [Dostęp: 04.03.2022]

⁴ Mazgal A., Tarkowski A., Nauka programowania w szkołach. Czas na upgrade – perspektywa 2014, Warszawa 2014, <http://centrumcyfrowe.pl/czytelnia/nauka-programowania-w-szkolach-czas-na-upgrade-%E2%80%92-perspektywa-2014/> [Dostęp: 07.03.2022]



Co w tym wszystkim najważniejsze, pisze Piotr Szawdyński, rozpoczynając naukę programowania, należy zdawać sobie sprawę, że to nie jest coś, co można opanować w kilka dni czy też tygodni. Realizację programowania należy rozpocząć od wyjaśnienia, przybliżenia znaczenia słowa „programowanie”⁵. Jak wskazuje Adam Boduch, z określeniem „nauczanie programowania” spotkaliśmy się nie raz. Z pewnością, dodaje Boduch, podczas oglądania telewizji, przeglądania prasy, a także w życiu codziennym. Niestety wiele osób w ogóle nie wie, co to pojęcie tak naprawdę oznacza. Na pewno, dodaje Boduch, zdarzają się zapaleńcy, ponieważ programowanie jest ciekawe, ale trzeba mieć predyspozycje do takiej pracy. Jeżeli ktoś zabiera się za programowanie z przymusu, to pewnie nie będzie miał z tego żadnej przyjemności. Jednak jeśli samodzielnie napisany program (choćby nie wiem jak był skromny) przyniesie swojemu twórcy poczucie satysfakcji i dumy — zapewne programowanie wciągnie go jeszcze bardziej — dodaje A. Boduch, który określa programowanie słowami: „Ogólnie mówiąc, polega na rozwiązywaniu pewnych problemów. Jest to dziedzina zajmująca się tworzeniem oprogramowania. (...) Celem programisty jest napisanie programu, który wspomocze człowieka przy wykonywaniu naprawdę trudnych obliczeń”⁶. Warto w tym miejscu dodać, że celem opracowania jest przede wszystkim przybliżenie Czytelnikowi praktycznych aspektów nauczania programowania z wykorzystaniem mikrokontrolerów. Kolejnym celem jest pokazanie, jak uczenie programowania w oparciu o idee otwartości, współpracy oraz zniesienie paradygmatu „mistrz-uczeń” przybliży nas do nowoczesnej szkoły. A więc takiej, która uczy rozwiązywania problemów, buduje kompetencje społeczne oraz przygotowuje młodzież do mierzenia się z wyzwaniami rynku pracy.

⁵ Szawdyński P., Podstawy programowania – teoria i terminologia, <http://cpp0x.pl/kursy/Kurs-C++/Poziom-1/Podstawy-programowania-teoria-i-terminologia/5> [Dostęp: 07.03.2022]

⁶ A. Boduch, Programowanie. Wstęp do nauki, http://4programmers.net/Z_pogranicza/FAQ/Czym_jest_programowanie. [Dostęp: 09.03.2022]



1. PROGRAMOWANIE – EDUKACYJNA INNOWACJA PRZYSZŁOŚCI

Programowanie jest rozumiane przeze mnie znacznie szerzej niż tylko samo napisanie programu w języku programowania. To cały proces, informatyczne podejście do rozwiązywania problemu: od specyfikacji problemu (określenie danych i wyników, a ogólniej – celów rozwiązania problemu), przez znalezienie i opracowanie rozwiązania, do zaprogramowania rozwiązania, przetestowania jego poprawności i ewentualnej korekty przy użyciu odpowiednio dobranej aplikacji lub języka programowania. Tak rozumiane programowanie powinno być częścią zajęć informatycznych od najmłodszych lat, wpływać na sposób nauczania innych przedmiotów, służyć właściwemu rozumieniu pojęć informatycznych i metod informatyki. Powinno wspomagać też kształcenie takich umiejętności jak: logiczne myślenie, precyzyjne prezentowanie myśli i pomysłów, sprzyja dobrej organizacji pracy, buduje kompetencje potrzebne do pracy zespołowej i efektywnej realizacji projektów. Umiejętności nabyte podczas programowania powinny być przydatne na zajęciach z innych przedmiotów, jak i później w różnych zawodach, niekoniecznie informatycznych.

Cele ogólne kształcenia informatycznego są takie same dla wszystkich etapów edukacyjnych. Opis wymagań szczegółowych ma charakter spiralny (przyrostowy) – na każdym etapie edukacyjnym wymaga się od uczniów umiejętności zdobytych na wcześniejszych etapach edukacyjnych i rozszerza się je o umiejętności nowe. Istnieje zatem silna potrzeba zmiany sposobu postrzegania przedmiotów informatycznych w szkole. Mówi o tym Bohumir Soukup, czeski programista i twórca narzędzi do nauki programowania (m.in. Baltie, adresowanego do dzieci). Twierdzi on, iż większość kompetencji informatycznych już teraz jest lub za chwilę się stanie czymś normalnym, co nie będzie wymagało nauki w szkole. Wynika to m.in. z kontaktu dzieci z nowymi technologiami w ich w środowisku wzrastania⁷. Stale zwiększający się udział technologii w życiu

⁷ Soukup B.: Nauka programowania zamiast przedmiotu informatyka w szkołach podstawowych. [W:] Człowiek - Media - Edukacja. Red. naukowa: J. Morbitzer, D. Morańska,



człowieka wzmaga potrzebę wykształcenia kadr, które będą w stanie rozwiązywać problemy technologiczne, z których jeszcze dzisiaj nie zdajemy sobie nawet sprawy.

Osobami, które będą determinantem zmian, ale także gwarantem technologicznego poprawnego wzrastania, są programiści. W ostatnich kilku latach zaczyna być dotkliwie odczuwalny (zarówno w Polsce, jak i na świecie) brak dobrych profesjonalistów z takimi kwalifikacjami, którzy stanowią siłę napędową rozwoju technologii w przyszłości. Jednak B. Soukup w samym programowaniu postrzega szersze znaczenie i proponuje, aby wszyscy programowali. Nie po to, aby wszyscy byli programistami, ale dlatego, iż dzięki tej umiejętności jest możliwe kształcenie kompetencji kluczowych człowieka XXI wieku. Będą one niezbędne dla informatyków, ale również dla ekonomistów, inżynierów, kierowników, czyli dla wszystkich osób, od których będzie się wymagało umiejętności rozwiązywania problemów z wykorzystaniem nowych technologii. Na rynku polskim stopniowo przybywa specjalistów w dziedzinie ICT, ale ich liczba pozostaje poniżej średniej dla UE. Specjaliści w dziedzinie ICT stanowią niższy odsetek siły roboczej (3,4%) w stosunku do średniej unijnej (4,3%), a zaledwie 15% pracujących kobiet jest zatrudnionych na tych stanowiskach zestawionych w tabeli 1⁸.

E. Musiał. Wydawca: Wyższa Szkoła Biznesu w Dąbrowie Górniczej, Dąbrowa Górnicza 2015

⁸ Komisja Europejska, raport DESI 2021, <https://cyberpolicy.nask.pl/indeks-gospodarki-cyfrowej-i-spoleszenstwa-cyfrowego-desi-2021/> [Dostęp: 14.03.2022]



Tabela 1. Kapitał ludzki

	Polska			UE
	DESI 2019	DESI 2020	DESI 2021	DESI 2021
1a1 Co najmniej podstawowe umiejętności cyfrowe % osób	46% 2017	44% 2019	44% 2019	56% 2019
1a2 Ponadpodstawowe umiejętności cyfrowe % osób	21% 2017	21% 2019	21% 2019	31% 2019
1a3 Co najmniej podstawowe umiejętności informatyczne % osób	49% 2017	46% 2019	46% 2019	58% 2019
1b1 Specjaliści w dziedzinie ICT % osób pracujących w wieku 15–74 lat	3,0% 2018	3,1% 2019	3,4% 2020	4,3% 2020
1b2 Kobiety-specjaliści w dziedzinie ICT % specjalistów w dziedzinie ICT	14% 2018	14% 2019	15% 2020	19% 2020
1b3 Przedsiębiorstwa zapewniające szkolenia z zakresu ICT % przedsiębiorstw	13% 2018	13% 2019	18% 2020	20% 2020
1b4 Absolwenci kierunków w dziedzinie ICT % absolwentów	3,5% 2017	3,8% 2018	3,8% 2019	3,9% 2019

Źródło: Komisja Europejska, raport DESI 2021, <https://cyberpolicy.nask.pl/indeks-gospodarki-cyfrowej-i-spolnoczenia-cyfrowego-desi-2021/>

Często wybieranym przez nauczycieli językiem w nauczaniu programowania, szczególnie w młodszych klasach, jest opracowany w MIT Media Lab dla celów edukacyjnych, wizualny język Scratch. Pomimo sceptycyzmu wielu osób do graficznego sposobu programowania, język ten jest jednym z najbardziej popularnych języków świata i według rankingu TIOBE, w lipcu 2020 zajmuje bardzo wysokie 17 miejsce. Ci, którzy jako główny język w nauce programowania wybrali Scratch, nie muszą rezygnować z poznawania systemów wbudowanych. Wiele popularnych platform daje taką możliwość, tym bardziej, że powstało dużo modyfikacji tego języka. W poszukiwaniu skutecznych strategii dydaktycznych współczesna szkoła sięga po coraz bardziej innowacyjne rozwiązania.

1.1 Rozwijanie myślenie komputacyjnego

W ostatnich latach analiza wyzwań stawianych przed obywatelami rozwijających się społeczeństw, w których gospodarka bazuje na wiedzy, doprowadziła do dalszego rozszerzenia niezbędnego zakresu kompetencji informatycznych o myślenie komputacyjne (ang. computational thinking), które obejmuje szeroki zakres intelektualnych narzędzi, reprezentujących spectrum



informatycznych metod modelowania i rozwiązywania problemów, znacznie szerszy niż tradycyjnie rozumiane myślenie algorytmiczne. W ślad za tą determinantą podstawa programowa z informatyki dla szkoły podstawowej wprowadziła pojęcie myślenia komputacyjnego, traktując je, obok umiejętności pisania, czytania i rachowania, jako podstawową umiejętność wymagającą alfabetyzacji. Pojęcie myślenia komputacyjnego jest nowe w założeniach programowych kształcenia ogólnego.

Warto zauważyć, że ogólnie umiejętność programowania to tylko część szeroko rozumianych kompetencji cyfrowych i medialnych, a powszechność technologii komunikacyjnych sprawia, że jakość życia obywateli współczesnego społeczeństwa w coraz większym stopniu zależy od zdolności rozumienia i korzystania z informacji docierających do nas za pośrednictwem mediów. Programowanie należy wprowadzić do edukacji formalnej i nieformalnej jak najwcześniej, od pierwszych klas szkół podstawowych, a nawet już w przedszkolu. Po odpowiedniej modyfikacji podstawy programowej przedmiotów informatycznych może być ona wdrażana we wszystkich klasach, szkołach oraz na wszystkich etapach kształcenia, czyli od edukacji wczesnoszkolnej, aż do matury. Prawdziwą alfabetyzacją XXI wieku jest programowanie, w sensie umiejętności korzystania z innowacyjnych możliwości technologii – komputerów, nie tylko z gotowych rozwiązań. W swoim opracowaniu M.M. Sysło pt. „Dość technologii, zaczynamy wszystkich uczniów uczyć informatyki i programowania”⁹ wskazuje także, że obecnie myślenie komputacyjne to informatyczne podejście do rzeczywistych problemów z różnych dziedzin. Wcześniej myślenie algorytmiczne, a aktualnie myślenie komputacyjne, integruje ludzkie myślenie z możliwościami komputerów, obok stosowania gotowych narzędzi i informacji kształtuje kreatywność w tworzeniu

⁹ M.M. Sysło, Dość technologii, zaczynamy wszystkich uczniów uczyć informatyki i programowania, http://www.kassk.pl/NTomysl_2014_MM Syslo.pdf [Dostęp: 17.04.2022]



własnych narzędzi i informacji. Wg Sysło¹⁰ myślenie komputacyjne określa użyteczne postawy i umiejętności, jakie każdy, nie tylko informatyk, powinien starać się wykształcić i stosować. Stanowi naturalne poszerzenie kompetencji określanych jako 3R (reading, writing, arithmetic), o umiejętności stosowania metod pochodzących z informatyki przy rozwiązywaniu problemów. Mające swoje korzenie w informatyce, myślenie komputacyjne daleko wykracza poza tradycyjną informatykę i stanowi ramy dla myślenia o problemach pochodzących z różnych dziedzin i metodach ich rozwiązywania. Autor powyższych słów, myślenie komputacyjne postrzega jako informatykę stosowaną w różnych dziedzinach wraz z możliwością nauczania wszystkich uczniów myślenia komputacyjnego – wykorzystanie doświadczenia z informatyki. Należy stosować przy tym, zdaniem M.M.Sysło, metodologię rozwiązywania problemów np. metodą projektów, bazującą na metodologii – zastosowana do realizacji zmienionej podstawy programowej kształcenia informatycznego. Umiejętności nabyte podczas programowania są również przydatne na zajęciach z innych przedmiotów, jak i później w różnych zawodach, niekoniecznie informatycznych. Umożliwiają uczniom przejście z pozycji cyfrowego konsumenta na pozycję cyfrowego twórcy oraz przyjęcie roli osoby władającej technologią, a nie tylko poddającej się jej. Nauczanie programowania w szkole nie jest więc jedynie kształceniem programistów – czytamy w „Mapie drogowej”¹¹ przygotowanej przez Radę ds. Informatyzacji Edukacji przy Ministrze Edukacji Narodowej.

1.2 TIK w nauczaniu informatyki

TIK (technologie informacyjno-komunikacyjne) są wszechobecne na zajęciach komputerowych i podczas nauki informatyki. Trudno wyobrazić sobie

¹⁰ M.M. Sysło, Myślenie komputacyjne. Nowe spojrzenie na kompetencje informatyczne, w: A.B. Kwiatkowska, M.M. Sysło (red.), Informatyka w edukacji. Informatyka dla wszystkich od najmłodszych lat, Toruń 2014, str. 15–32

¹¹ <http://szkoly-geoserver.itl.waw.pl/>. [Dostęp: 07.04.2022].



edukację w tych dziedzinach bez nowoczesnych technologii. Poza uczeniem się podstaw budowy komputera, uczniowie poznają zasady bezpiecznego korzystania z internetu, w tym aspekty etyczne, prawne i ochronę danych osobowych, komunikują się w cyberprzestrzeni, wyszukują, selekcjonują i gromadzą informacje, korzystają z wielu programów użytkowych. Tworzą i przekształcają teksty, grafikę, dźwięk, prezentacje multimedialne, dane liczbowe, wykonują złożone projekty, opracowują strony internetowe, rozwijają swoje zainteresowania. Te umiejętności stanowią podstawę wykorzystania TIK także na innych lekcjach oraz w życiu pozaszkolnym. Nauczyciele prowadzący zajęcia techniczne skupiają się głównie na tym, aby uczniowie rozumieli instrukcje obsługi i zasady działania urządzeń znajdujących się w ich otoczeniu, znali różne materiały konstrukcyjne i potrafili wykorzystać ich właściwości do projektowania, wykonywali szkice i rysunki techniczne, konstruowali modele z elementów, bezpiecznie uczestniczyli w ruchu drogowym.

Kształcenie informatyczne odbywa się głównie w kontakcie z wirtualnymi przedmiotami i informacjami na ich temat. Uczniowie przetwarzają materiały i wykonują z nich projekty, korzystają z różnych narzędzi i oprogramowania. To bezpośrednie obcowanie z urządzeniami i aplikacjami jest główną zaletą szkolnej edukacji technicznej i informatycznej. Można ułatwić uczniom kształcenie techniczne poprzez włączenie TIK do tego procesu. Dostęp do instrukcji obsługi online, nauka rzutowania i szybkie opracowywanie schematów w specjalnych programach, obserwowanie modeli przestrzennych i ich konstruowanie z elementów – w tym wszystkim pomoże komputer z odpowiednim oprogramowaniem i internetem.

Kluczem w rozwijaniu kreatywności, przedsiębiorczości i kompetencji cyfrowych uczniów jest bezpieczne i celowe wykorzystywanie technologii informacyjno-komunikacyjnych w realizacji podstawy programowej kształcenia ogólnego. Jak rozpoznać, że użycie TIK na lekcji zwiększa szanse osiągnięcia jej celów? Można przyjąć następujące wskaźniki: Uczniowie korzystający z TIK na lekcji:

- koncentrują się na celach uczenia się, a nie narzędziach,



- szybciej osiągają cele,
- ich poziom wiedzy, umiejętności i świadomości jest wyższy,
- są bardziej aktywni intelektualnie i zaangażowani w uczenie się.

Narzędzia technologii są intrygującymi gadżetami, ale za wyznaczanie i osiągnięcie przez wychowanków celów edukacji odpowiedzialny jest nauczyciel będący profesjonalistą w swojej dziedzinie. Samodzielność i kreatywność uczniów rozwijamy, zachęcając ich do działania, stawiając przed nimi zadania, które mają odpowiedni poziom trudności, pozwalając im popełniać błędy, brać odpowiedzialność za podejmowane decyzje. Technologie mogą być pomocne, ale nie zastąpią pierwiastka ludzkiego, bo tylko człowiek jest zdolny do wychodzenia poza schematy i nieszablonowego myślenia, a na tym opiera się kreatywność.

1.3 Metoda projektu jako strategia uczenia się

Współczesna dydaktyka przywiązuje dużą wagę do rozwoju kompetencji kluczowych u uczniów (łączyjących wiedzę, umiejętności i postawy odpowiednie do sytuacji). Właśnie nauczyciel jest i musi, nie tylko może, być odpowiedzialny za tego, za kogo przyjął z całą świadomością odpowiedzialność, czyli za ucznia. Powinien dobrze przygotować go do wyzwań XXI wieku. Aby zrealizować zadania wychowawcze i dydaktyczne, musi zastanowić się nad wyborem metod i narzędzi pracy atrakcyjnych dla wychowanków i dających satysfakcję osobistą. Powinien przyświecać mu cel: „Nie nudzić na lekcji siebie i uczniów”. Jak to zrobić? Wybrać innowacyjne metody i narzędzia pracy. Ciągle doskonalić swoje kompetencje. Nadążać za rozwojem nauki i techniki oraz zmianami zachodzącymi w społeczeństwie. naturalne jest, że i szkoły szukają w tym dynamicznie zmieniającym się świecie nowej formuły istnienia, nowej tożsamości, nowych organizacyjnych kształtów. Coraz częściej nauczyciele odchodzą od tradycyjnych metod (np. wykład, pogadanka, opis, opowiadanie, prelekcja, praca z książką), a wybierają innowacyjne metody: projekt edukacyjny, np. WebQuest, portfolio lub ePortfolio, drzewko decyzyjne, mapa mentalna, różne odmiany



dyskusji, metaplan, gry dydaktyczne, metoda projektu itp. Do wymienionych powyżej metod można zastosować odpowiednie narzędzia interaktywne. Te metody i narzędzia można ze sobą łączyć i przekształcać. Dzięki nim zajęcia są interesujące, wyzwalają w uczniach otwartość i kreatywność.

Metoda projektowa jest to metoda interdyscyplinarna, która zakłada u uczących się znaczną samodzielność i odpowiedzialność. Projekty mogą być wykonywane grupowo lub indywidualnie. Dzieci podczas pracy nad swoimi projektami powinny mieć zapewniony swobodny dostęp do różnych przedmiotów oraz pomocy nauczyciela. Metoda projektu pomaga stworzyć zintegrowany zespół, w którym uczniowie przez pracę w grupie uczą się rozwiązywania problemów, aktywnego słuchania, skutecznego komunikowania się, a także zdobywają poczucie własnej wartości. Metoda projektu wdraża dzieci do planowania oraz organizowania swojej pracy, a także do dokonywania samooceny. Podczas projektu praca odbywa się w indywidualnie ustalonym czasie i tempie. Metoda taka umożliwia doświadczenie solidarności oraz radości ze wspólnego wysiłku, a także poczucie identyfikacji z zadaniem, które pozwala przezwyciężyć konflikty między jednostkami. Projekt musi zawierać 4 podstawowe cechy:

1. Nabywanie wiedzy powinno zachodzić głównie dzięki samodzielnemu rozumowaniu i rozwiązywaniu problemów, a nie za pomocą pamięciowego przyswajania wiedzy.
 2. Aktywność dziecka ma na celu nie tylko gromadzenie wiedzy, czy doświadczeń, lecz także zmianę nastawienia dziecka do uczenia się.
 3. Dzieci pracują nad zagadnieniami, które ich interesują i wynikają z ich naturalnych potrzeb.
 4. Wprowadzenie rozważań teoretycznych następuje w miarę potrzeb i zawsze wynika z realizowanych działań praktycznych.
- Aby efektywnie pracować metodą projektu, należy przestrzegać określonych faz pracy, do których należą:

- zainicjowanie projektu,



- spisanie kontraktu,
- wybór tematu,
- podział na grupy,
- sformułowanie ogólnych oraz szczegółowych celów projektu,
- przygotowanie harmonogramu, podział pracy,
- eerealizacja projektu,
- prezentacja wyników projektu,
- ewaluacja.



2. PROGRAMOWANIE MIKROKONTROLERÓW

Tradycyjne metody nauki programowania są w większości zorientowane tekstowo, i nie chodzi tu tylko o samą czynność kodowania, lecz także o komunikację działający program – użytkownik. Dla dojrzałego ucznia taka konsolowa metoda, oparta na tak zwanym standardowym wejściu i wyjściu, może być ciekawa i wystarczająca, ale część uczniów oczekiwałaby możliwości co najmniej graficznych. Ponieważ jest to tylko kłopotliwym dodatkiem, bowiem wymaga dużo nakładu pracy, a nie stanowi istoty nauczanych elementów języka, w trakcie nauczania linia poleceń pozostaje podstawowym narzędziem komunikacji.

Systemy wbudowane, szczególnie w ich przyjaznym współczesnym wydaniu, mogą pozwolić na naukę najważniejszych konstrukcji programistycznych, wychodząc przy tym poza standardowe wejście i wyjście. Dają one, nieosiągalną dotychczas dla nieprofesjonalistów, możliwość interakcji pomiędzy aplikacją a światem fizycznym. Pozwalają nawet niedoświadczonym programistom na sterowanie pracą robotów.

Umiejętność doboru odpowiedniego środowiska informatycznego w realizacji rozwiązań problemów jest kluczową umiejętnością współczesnego programisty. Obecne możliwości łączenia mikroelektroniki i programowania na tanich i ogólnodostępnych platformach, są olbrzymią szansą uatrakcyjnienia procesu edukacyjnego nauki programowania. W kolejnych rozdziałach poruszam praktyczne aspekty programowania mikrokontrolerów Micro:bit oraz Arduino. We wszystkich częściach duży nacisk położono na ekspresję twórczą dzieci. Przedstawione programy, dobór wprowadzanych umiejętności oraz rodzaje zadań mają prowadzić uczniów od czynności odtwórczych do twórczości. Uczniowie tworzą na komputerze autorskie programy, rozwijają w ten sposób swoje zainteresowania i pasje.

Uczniowie ze specjalnymi potrzebami edukacyjnymi mogą systematyzować swoją wiedzę dzięki rozwiązywaniu zadań wymagających tylko odtworzenia prezentowanych treści. Szczególnie uzdolnieni lub zainteresowani mają



możliwość rozwijania swoich pasji przez rozbudowę projektu, polegającą na budowie makiet drogowych z wykorzystaniem np. technologii druku 3D. W aspekcie indywidualnych potrzeb edukacyjnych najważniejsza jest strategia, którą przyjmie nauczyciel dla konkretnego ucznia, oraz dobór dostosowań w szczególności pod kątem organizacji pracy, form i metod nauczania - uczenia się.

2.1 Charakterystyka platformy Micro:bit

Bardzo dobrze do graficznego trybu programowania jest przystosowana platforma Micro Bit (nazwa stylizowana na micro:bit), opracowana przez BBC dla celów edukacyjnych. Posiada dedykowane środowisko programistyczne dostępne online oraz dużą międzynarodową społeczność. Główne metody programowania tego systemu wbudowanego opierają się na językach Python, JavaScript i podobnym do Scratcha języku Blocks. Przeznaczenie edukacyjne skutkuje bardzo łatwą obsługą całego systemu. Niewielką niedogodnością może być natomiast bardzo mała ilość dokumentacji w języku polskim. Micro:bit nie posiada nawet polskiej strony w Wikipedii. Oficjalna strona projektu microbit.org w języku angielskim jest za to niezwykle bogata i została już przetłumaczona na kilkanaście popularnych języków.

Nauka podłączania zewnętrznych modułów do micro:bit na przykładzie diod led oraz kontrolowanie czasu ich działania układu stanowi doskonałe połączenie mikroelektroniki z nauką programowania.

Pomimo, że prezentowane w niniejszej publikacji, zewnętrzne elementy modułu BBC Micro Bit są wystarczające do realizacji nawet bardzo skomplikowanych projektów, to czasami warto rozważyć możliwość rozszerzenia funkcjonalności modułu o dodatkowe lub lepsze czujniki. Także komunikacja z użytkownikiem przy pomocy matrycy LED o wymiarach 5 × 5 diod, pomimo że pozwala na wyświetlenie dowolnego tekstu, nie może być uznana za wydajną i estetyczną. Jeżeli moduł ma wyprowadzać dużo danych dla użytkownika, warto rozważyć podłączenie zewnętrznego wyświetlacza. Jedną z najważniejszych grup modułów rozszerzeń są ekspandery wyprowadzeń. Ułatwiają one dostęp do



fizycznych styków (pins). W micro:bitcie w prosty sposób można podłączyć się jedynie do 3 pinów przy pomocy „krokodylków” lub wtyków bananowych. Rodzajów ekspanderów jest dosyć dużo, dlatego należy dokonać wyboru na podstawie analizy konkretnego projektu. Asortyment dodatków do micro:bita jest bardzo szeroki i obejmuje między innymi:

- ekspandery wyprowadzeń,
- dodatkowe czujniki – zarówno pojedyncze, jak i zestawy,
- sterowniki silników i zewnętrzne przekaźniki,
- moduły Wi-Fi,
- różnego rodzaju wyłączniki, gamepady, a nawet proste konsole gier,
- mikrofony, beepery, głośniki,
- wyświetlacze i panele LED,
- platformy robotów, i inne.

2.2 Programowanie z platformą Arduino

Arduino jest to płytką uruchomieniową, czyli służy do uruchamiania elementów elektronicznych, za jej pomocą możemy odczytywać poziomy napięcia, czy też impulsy elektryczne, jak ma to miejsce w różnego rodzaju czujnikach. Możemy też wysłać impulsy elektryczne w celu sterowania innymi urządzeniami i są to wszelkiego rodzaju wyświetlacze, diody, czy silniki. Na potrzeby tego opracowania wykorzystałem Arduino Uno R3, jest to najpopularniejszy model z całej serii, ale też najlepszy dla osób, które zaczynają swoją przygodę z platformą. Uno wyposażone jest we wbudowany programator, dzięki czemu nie musimy martwić się o zakup dodatkowego urządzenia, a kod wgrywamy na płytkę przez kabel USB. Kolejną zaletą jest demontowalny mikrokontroler, który możemy w razie awarii wymienić, bez konieczności zakupu nowej płytki.



2.2.1 Zalety i możliwości wykorzystania Arduino

Zaletami platformy jest niewątpliwie łatwość obsługi, dzięki czemu możemy bardzo szybko nauczyć się, jak korzystać z platformy. Ponadto, ilość dostępnych urządzeń, które możemy podłączyć do płytki jest niezliczona. Nic także nie stoi na przeszkodzie, aby samemu budować czujniki i łączyć je z płytką, co daje duże pole do popisu dla osób, które lubią poznawać działanie każdego urządzenia od początku. Największym, moim zdaniem, plusem dla Arduino jest ogromna społeczność. Fora, kanały na YouTube, czy też grupy na portalach społecznościowych chętnie wyciągają pomocną dłoń w stronę początkujących.

Możliwości zastosowania Arduino jest tak wiele, że ciężko by wymienić chociaż 1% z nich. Tak naprawdę każde urządzenie, które mamy w domu jesteśmy w stanie włączyć, korzystając z Arduino. Nic nie będzie stało na przeszkodzie, jeśli będziemy chcieli, aby zbudować urządzenie z dostępem do internetu.

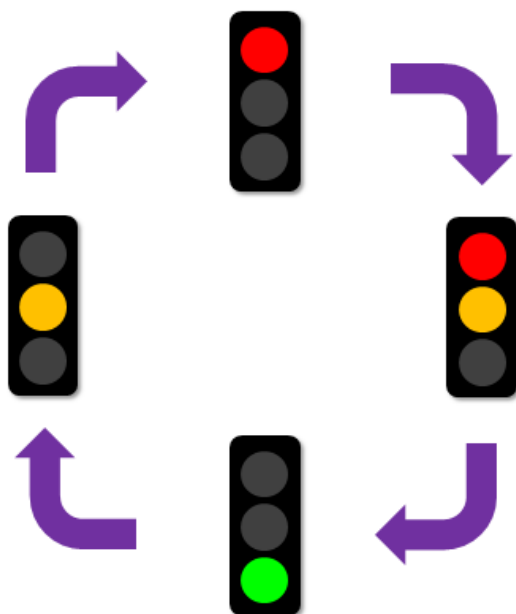
Zakres możliwości płytki zaczyna się na sterowaniu prostymi elementami elektronicznymi takimi jak diody, a kończy się na niezwykle precyzyjnych urządzeniach CNC, takich jak frezarki, czy drukarki 3D. Jeśli nie chcemy tworzyć od zera swojego projektu, możemy skorzystać z wielu serwisów, w których użytkownicy opisują swoje projekty krok po kroku.



3 SYMULATOR ŚWIATEŁ ULICZNYCH

Celem prezentowanego projektu realizowanego z mikrokontrolerami jest pokazanie, że programowanie może być bardzo ciekawym zajęciem, a podstawy elektroniki nie muszą być trudne. W związku z tym, cały projekt został przygotowany tak, aby zapewnić realizację projektu z wykorzystaniem dodatkowych materiałów. Decyzję dotyczącą wyboru platformy oraz środowiska programowania podejmie nauczyciel na podstawie możliwości realizacji zajęć, czasu lub po konsultacji z uczniami.

Światła drogowe wykonane za pomocą mikrokontrolerów działają na tej samej zasadzie co rzeczywiste światła drogowe, więc gdy samochody mają czerwone światło to pieszym na tej drodze włącza się zielone światło i analogicznie - jak samochody mają zielone światło, to piesi czerwone. Światła drogowe dla samochodów mają kolory zielony, żółty i czerwony. Gdy świeci się zielone światło to można jechać, żółte światło oznacza, że zaraz zapali się czerwone światło, które zabrania wjazdu na skrzyżowanie. Jeśli świeci się światło czerwone i żółte to oznacza, że za chwile zapali się światło zielone, które oznacza, że można wjechać za sygnalizator, lub przejść na przejściu dla pieszych.



Rysunek 1. Sekwencja światel drogowych, źródło: opracowanie własne



Poniżej przedstawiłem przykładowe ćwiczenia do wykonania zgodnie z wcześniej zaprezentowaną logiką. Ćwiczenia mają charakter wprowadzenia do symulacji rzeczywistych układów świateł drogowych.

Sygnały podstawowe:

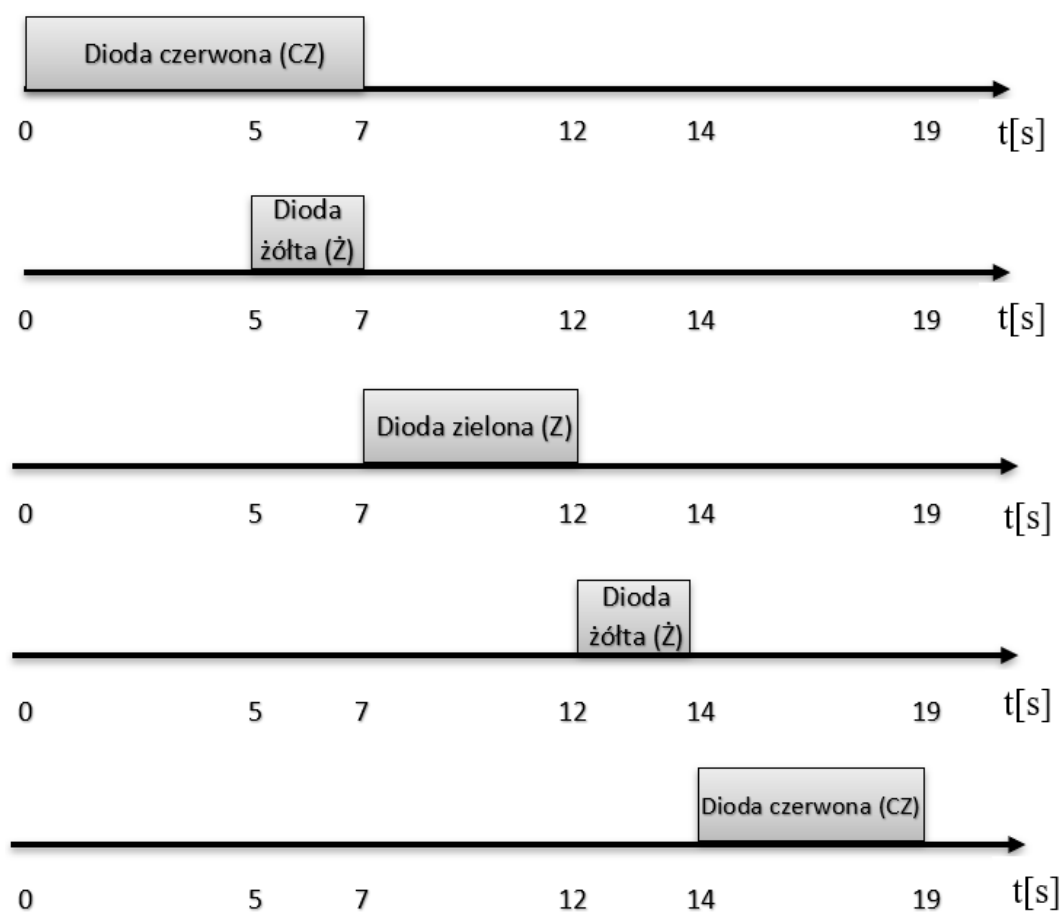
- 1) sygnał czerwony – zabrania wjazdu za sygnalizator, a pieszym – wejścia na przejście dla pieszych – dioda czerwona (CZ) zapala się na 5 s.,
- 2) sygnał czerwony z żółtym – zabrania wjazdu za sygnalizator informując jednocześnie o zmianie nadawanego sygnału na zielony – dioda czerwona (CZ) zapala się na 5 s., po tym czasie dioda żółta oraz zielona zapalają się na kolejne 2 s.,
- 3) sygnał zielony – zezwala na wjazd za sygnalizator, a pieszym – wejście na przejście dla pieszych – dioda zielona (Z) zapala się na 5 s.,
- 4) sygnał żółty ciągły – zabrania wjazdu za sygnalizator, chyba że w momencie jego zaświecenia kierujący pojazdem był tak blisko sygnalizatora, że nie mógłby się przed nim zatrzymać bez gwałtownego hamowania, informując jednocześnie o zmianie nadawanego sygnału na czerwony – dioda żółta (Ż) zapala się na 2 s.

Pozostałe sygnały:

- 1) sygnał zielony pulsujący – poleca pieszym znajdującym się na przejściu dla pieszych jak najszybciej je opuścić, informując jednocześnie o zmianie nadawanego sygnału na czerwony – dioda zielona (Z) zapala się i gaśnie z częstotliwością 1 s. przez 5 s.
- 2) sygnał żółty pulsujący – ostrzega o zwiększonym ryzyku niebezpieczeństwa oraz o zmianie organizacji ruchu. W tej sytuacji należy stosować się do znaków drogowych, a w przypadku ich braku, odwołać się do kodeksu ruchu drogowego – dioda żółta (Ż) zapala się i gaśnie z częstotliwością 1 s. przez dowolny czas.



Rysunek 2. Przykład sygnalizacji świetlnej z sekundnikiem, źródło: [link do lovekrakow.pl](http://link.do/lovekrakow.pl)



Rysunek 3. Schemat sterowania sygnalizacją świetlną, źródło: opracowanie własne



3.1 Wykaz elementów układu symulatora

W tym podrozdziale zaprezentuję listę potrzebnych do stworzenia obwodu symulatora sygnalizacji świetlnej elementów.

Elektronikę można podzielić na cyfrową oraz analogową. Cyfrowa ogranicza się do tego, że dane wejście lub wyjście może być w jednym z dwóch stanów.

Wysokim (1) lub niskim (0). W praktyce oznacza to, że Arduino rozpozna na swoim wejściu 0V lub 5V (w uproszczeniu).

Praktycznie nie da się poprawnie kontrolować pracy diody poprzez ustalenie na niej określonego napięcia. Musiałoby ono podążać za zmianami temperatury i zmianami struktury, co nie jest proste. Dlatego stosuje się zasilanie diody LED określonym, stałym prądem. Do zasilania LEDa wystarczy źródło zasilania oraz element ograniczający prąd, czyli rezystor.

Celem jest stworzenie obwodu z zastosowaniem mikrokontrolera i trzech diod LED: czerwonej, żółtej i zielonej. Wykorzystując te diody oraz płytke Arduino wykonamy „małą” sygnalizację świetlną.

Wykorzystane części:

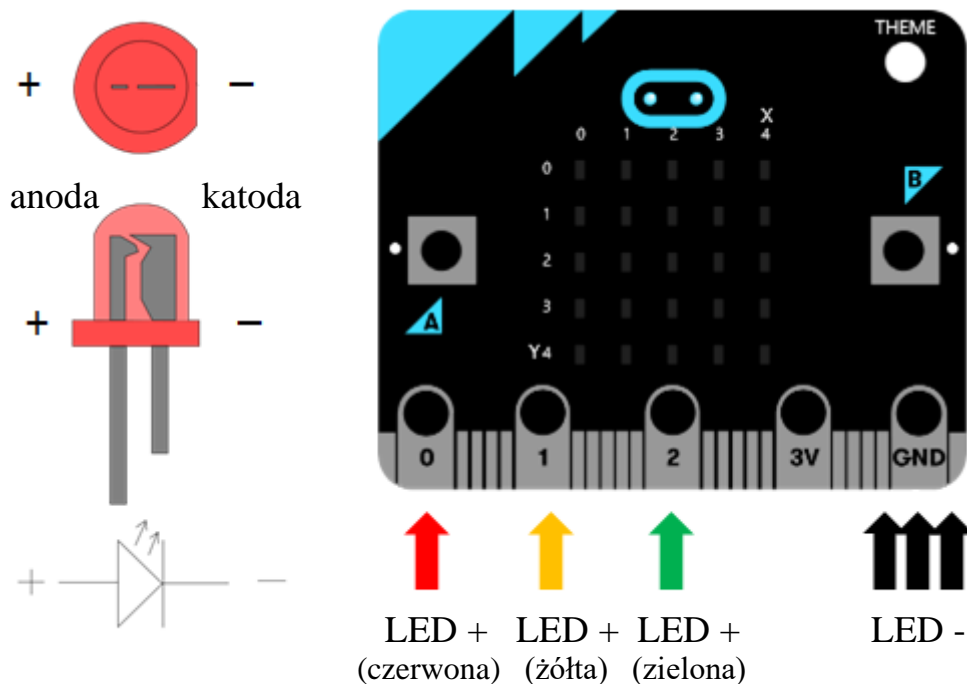
- mikrokontroler BBC Micro:bit/Arduino UNO,
- trzy diody led w kolorach czerwonym, żółtym i zielonym,
- trzy rezystory 220 Ω ,
- płytka stykowa np. 400 otworów,
- przewody połączeniowe,
- przewód USB.

Do zasilania układu wykorzystamy port USB. Wystarczy połączenie komputera z Arduino poprzez kabel USB (ten sam, który używany jest podczas programowania).

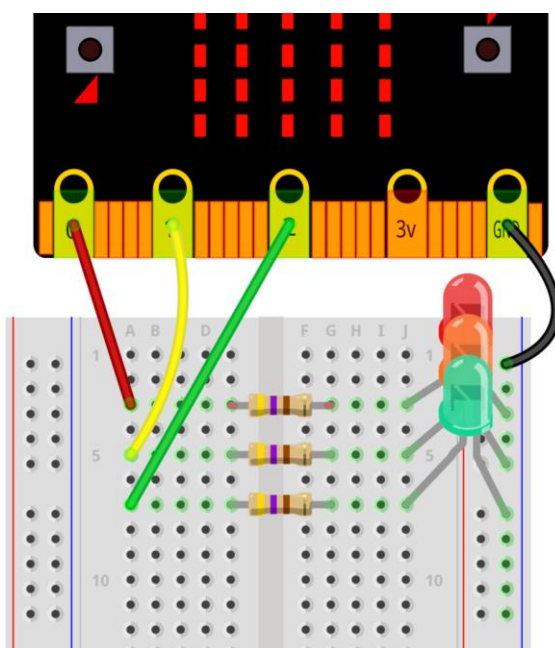


3.2 Podłączenie i programowanie modułu Micro:bit

Na rysunku przedstawiono rozmieszczenie podstawowych, ważnych z punktu widzenia użytkownika, komponentów modułu micro:bit.



Rysunek 4. Sposób podłączenia diod led do płytki micro:bit, źródło: opracowanie własne



Rysunek 5. Montaż elementów na płytce stykowej, źródło: opracowanie własne



Poniższy program stanowi powtarzającą się sekwencją występujących po sobie kolejno zdarzeń, polegających na zaświeceniu diod LED zgodnie z zasadą działania sygnalizatora świetlnego.

```
na zawsze
  zapis cyfrowy pin P0 na 1
  pauza (ms) 5000
  zapis cyfrowy pin P1 na 1
  pauza (ms) 2000
  zapis cyfrowy pin P0 na 0
  zapis cyfrowy pin P1 na 0
  zapis cyfrowy pin P2 na 1
  pauza (ms) 5000
  zapis cyfrowy pin P2 na 0
  zapis cyfrowy pin P1 na 1
  pauza (ms) 2000
  zapis cyfrowy pin P1 na 0
```

Rysunek 6. Program symulatora świateł drogowych w języku blokowym, źródło: opracowanie własne

Rozszerzeniem tej sekwencji będzie wykorzystanie matrycy świetlnej LED mikrokontrolera micro:bit jako sekundnika do wyświetlania czasu potrzebnego do zmiany świateł symulatora. W tym celu utworzona została zmienna „sekundnik”, której zadaniem jest przechowywanie i wyświetlanie czasu w sekundach.



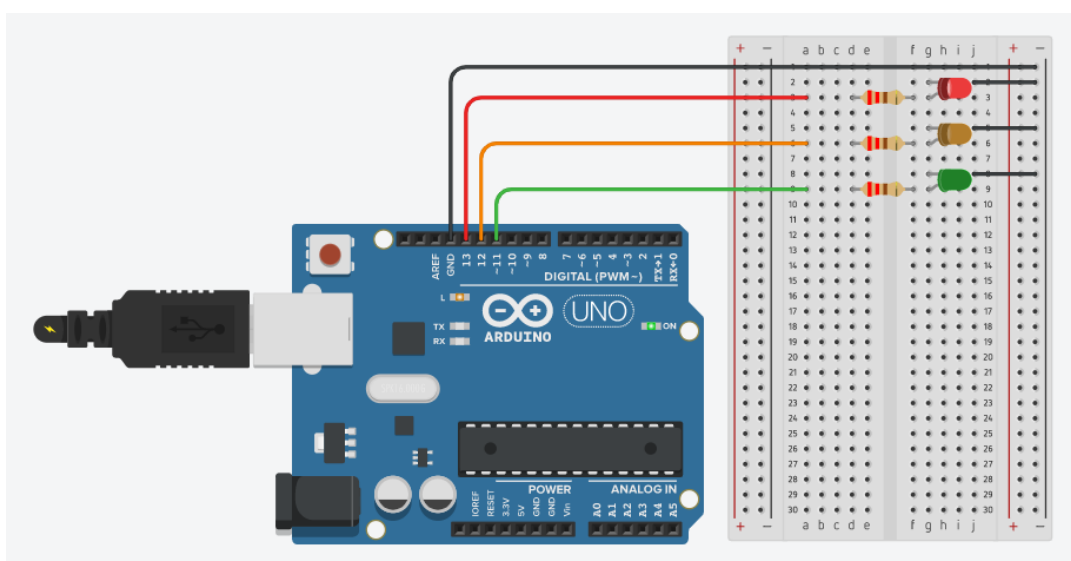
```
na zawsze
  pokaż LED
  zapis cyfrowy pin P0 na 1
  ustaw sekundnik na 5
  powtórz 5 razy
  wykonaj
    pokaż numer sekundnik
    pauza (ms) 1000
    zmień sekundnik o -1
  zapis cyfrowy pin P1 na 1
  pauza (ms) 2000
  pokaż LED
  zapis cyfrowy pin P0 na 0
  zapis cyfrowy pin P1 na 0
  zapis cyfrowy pin P2 na 1
  ustaw sekundnik na 5
  powtórz 5 razy
  wykonaj
    pokaż numer sekundnik
    pauza (ms) 1000
    zmień sekundnik o -1
  zapis cyfrowy pin P2 na 0
  zapis cyfrowy pin P1 na 1
  pauza (ms) 2000
  zapis cyfrowy pin P1 na 0
```

Rysunek 7. Program symulatora świateł drogowych z sekundnikiem w języku blokowym, źródło: opracowanie własne



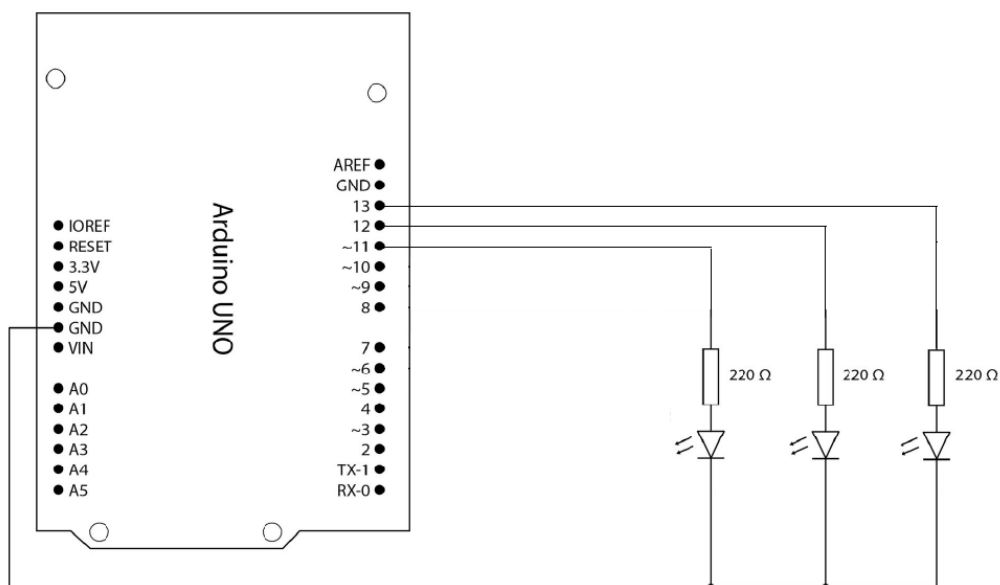
3.3 Podłączenie i programowanie obwodu z Arduino

Realizację projektu rozpoczynamy od podłączenia czerwonej diody LED. Dobra wartość rezystancji zabezpieczającym diodę przed zbyt wysokim prądem wynosi $220\ \Omega$. Na płytce stykowej podłączamy kolejno żółtą oraz zieloną diodę świecące LED przez rezystory. Jak widać na schemacie 7, przewody zasilające są podłączone do pinów płytki. Następnie poprzez rezystory do anod diod.



Rysunek 8. Schemat podłączenia układu do Arduino, źródło: opracowanie własne

Jeśli chodzi o podłączenie do pinów, z których korzystamy do zasilania diody, do sterownika, możemy je dowolnie zmienić.



Rysunek 9. Schemat elektryczny podłączenia diod led do Arduino, źródło: opracowanie własne

Układ będziemy programować jedynie przy użyciu portu USB i środowiska ARDUINO IDE. Język Arduino IDE jest zbliżony do C++, więc trzeba znać podstawy tego języka, aby tworzyć programy w IDE. Dostępnych jest wiele dodatkowych bibliotek do Arduino IDE, tworzonych na licencji oprogramowania open source przez tysiące hobbystów.

Przepisując lub kopiując program do IDE uruchamiamy kompilację.

Jeśli kompilator wykrył jakieś błędy, pole na dole będzie zawierało czerwone wpisy, jeśli wszystko pójdzie dobrze na czarnym polu, pojawi się komunikat białymi znakami o zajętej pamięci.

Język Arduino ma dwa podstawowe bloki:

```
void setup() {  
}  
void loop() {  
}
```



Gdy program się włącza najpierw, odczytuje funkcje setup i pobiera z niej podstawowe dane i wartości początkowe, na których później program pracuje.

W naszym programie w bloku setup mamy:

```
pinMode (kolor led, OUTPUT);
```

Instrukcja pinmode ustala działanie pinu albo na wejście INPUT, albo na wyjście OUTPUT. Przy INPUT sygnały są pobierane, przy OUTPUT sygnały są wysyłane do podłączonego urządzenia, w tym wypadku jest to dioda LED.

Drugi z bloków to funkcja loop (ang. powtarzaj). Funkcja loop jest niczym innym jak nieskończoną pętlą, która jest podstawową instrukcją wielu języków programistycznych. Pętla biegnie od początku funkcji loop, dociera do końca i znów wraca na początek i rozpoczyna kolejny obieg.

W funkcji loop mamy:

```
digitalWrite (czerwony, HIGH);
```

```
delay (5000);
```

- 1) `digitalwrite (PIN, STATE)` - instrukcja ta nadaje stan wysoki, lub niski wyjściu PIN, czyli mamy na 7 pinie stan wysoki, czyli płynie przez niego prąd.
- 2) `delay (time milliseconds)` - bardzo prosta instrukcja, która nie robi nic innego, jak zatrzymuje działanie programu na podaną liczbę milisekund. W naszym przypadku program staje na 2 lub 5 sekund w zależności od wywołanej instrukcji `delay`.

Na początek przepuszczamy prąd do diody czerwonej. Po 5 sekundach program załącza diodę żółtą. Po kolejnych 2 sekundach wyłączamy prąd diod czerwonej i żółtej, a następnie załączamy diodę zieloną na 5 sekund. Kolejno świeci się dioda żółta przez 2 sekundy. Obieg powtarza się i trwa, dopóki nie wyłączymy układu. Poniżej program sterowania w środowisku Arduino ID.



```
int czerwony = 13;

int zolty = 12;

int zielony = 11;

void setup() {

  pinMode(czerwony, OUTPUT);

  pinMode(zolty, OUTPUT);

  pinMode(zielony, OUTPUT);

}

void loop() {

  // włączone czerwone światło przez 5 sekund

  digitalWrite(zielony, LOW);

  digitalWrite(zolty, LOW);

  digitalWrite(czerwony, HIGH);

  delay(5000);

  // włączone żółte i czerwone światło przez 2 sekundy

  digitalWrite(zielony, LOW);

  digitalWrite(zolty, HIGH);

  digitalWrite(czerwony, HIGH);

  delay(2000);

  // włączone zielone światło przez 5 sekundy

  digitalWrite(zielony, HIGH);

  digitalWrite(zolty, LOW);

  digitalWrite(czerwony, LOW);

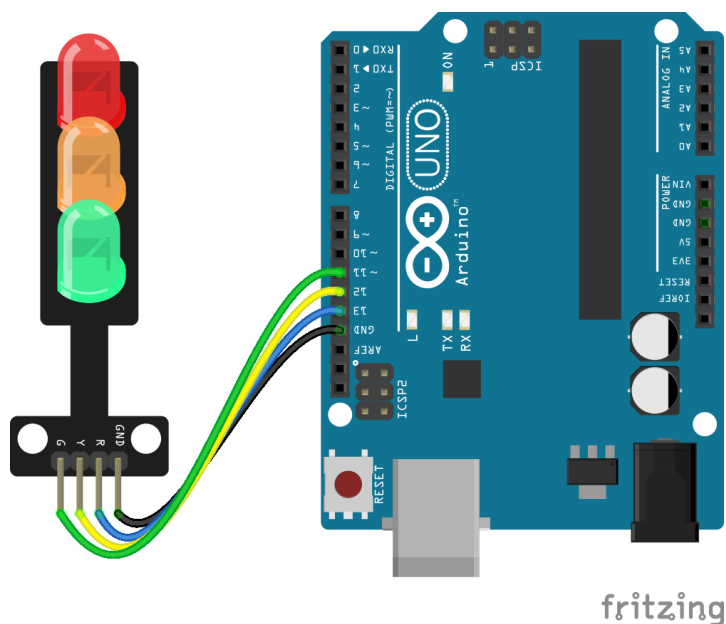
  delay(5000);

  // włączone żółte światło przez 2 sekundy
```



```
digitalWrite(zielony, LOW);  
  
digitalWrite(zolty, HIGH);  
  
digitalWrite(czerwony, LOW);  
  
delay(2000);  
  
}
```

Identyczny program pozwoli nam sterować modułem sygnalizatora świetlnego drogowego przedstawionym na rysunku 9. Źródłem światła są tu trzy 8-milimetrowe diody LED połączone katodami (wspólne GND diod LED). Moduł pozwala w prosty sposób zbudować makietę z aktywną sygnalizacją świetlną. Warto zauważyć, że moduł nie wymaga podłączenia rezystorów, które zostały już wlutowane w układ sygnalizatora.



Rysunek 10. Schemat podłączenia modułu sygnalizatora świetlnego do Arduino, źródło: opracowanie własne



PODSUMOWANIE

Wszystkie prezentowane projekty nie stanowią eksperymentów myślowych, ale zostały fizycznie zmontowane podczas tworzenia skryptów. Wszystkie zostały przetestowane i działały prawidłowo, czego dowodem jest dokumentacja zdjęciowa. Trzeba jednak mieć świadomość, że konkretna instalacja i konfiguracja, a także ciągle aktualizowane oprogramowanie może sprawić, że nie wszystko zadziała „od ręki”. Nie należy się jednak zniechęcać, bo poszukiwanie błędów i nowych rozwiązań może być równie fascynujące, co samo tworzenie.

Wszystkie rysunki, schematy i programy są mojego autorstwa, otrzymane przy pomocy otwartych standardów.

Z satysfakcją należy odnotować, znajdujący się w podstawie programowej zapis ustawodawcy o różnorodnych środowiskach informatycznych. Do tej pory bowiem, nauka informatyki, zwłaszcza na poziomie podstawowym, oscylowała pomiędzy klawiaturą a ekranem monitora, zawężając proces edukacyjny do mało interesujących, coraz bardziej odstających od zmieniającej się rzeczywistości zagadnień. Współcześnie elementy informatyki odnajdziemy już w nieomal każdej dziedzinie działalności ludzkiej i znakiem czasu jest omówienie na poziomie szkoły, specyfiki informatyki w szerszym zakresie. Przełomowym jest zapis o stosowaniu informatyki w elementach robotyki.

Wychodząc naprzeciw nowej podstawie programowej zaprezentowałem możliwości zastosowania platform programistycznych micro:bit oraz Arduino w procesie dydaktycznym. Otwierają one nowe możliwości stosowania robotyki, a zarazem bardziej zaawansowanych układów elektronicznych.

Celem projektu było przygotowanie narzędzi – ogólnodostępnych, tanich, niewymagających dużych nakładów finansowych, elastycznych i łatwych w użyciu w szkole. Język programowania Arduino jest oparty na środowisku Wiring i zasadniczo na języku C/C++5, stąd też następuje naturalne przejście od teoretycznej nauki programowania do programowania układów zewnętrznych, poza komputerem PC.



Szczególnie obejrzenie efektów swojej pracy programistycznej poza monitorem komputera jest nader atrakcyjną formą dla ucznia. Należy pamiętać o tym aspekcie, dobierając sekwencję treści w nauczaniu programowania.

Nauka programowania nie jest zarezerwowana wyłącznie dla chcących pracować w przyszłości z komputerem. Każdy, kto się go uczy, nabywa umiejętności techniczne oraz wiele innych, takich jak: umiejętność precyzyjnego i logicznego myślenia, rozwijania kreatywności i pomysłowości, ćwiczenie umiejętności dbanie o szczegóły, poznawanie nowych sposobów rozwiązywania problemów. Obowiązek rozwijania u uczniów myślenia komputacyjnego spoczywa po części na wszystkich nauczycielach sięgających po narzędzia technologii informacyjno-komunikacyjnych (TIK). Jednak szczególnie dotyczy on nauczycieli informatyki. Związane jest to z posiadanym poziomem kompetencji cyfrowych, narzędzi, które wykorzystują nauczyciele w swojej pracy oraz cech i funkcji, które powinien spełniać wzorcowy model narzędzia okołoinformatycznego, które, w moim przekonaniu, będzie służyło kształtowaniu myślenia komputacyjnego u uczniów.



BIBLIOGRAFIA Z UWZGLĘDNIENIEM NETOGRAFII

Kwiatkowski, S.M. (2018). *Kompetencje przyszłości*. Warszawa: Fundacja Rozwoju Systemu Edukacji

Podstawa programowa (2017). *Szkoła podstawowa IV-VIII* [online]. Dostępny w: <https://podstawaprogramowa.pl/Szkola-podstawowa-IV-VIII> [Dostęp 25.04.2022]

Janczak, D. (2018). *TIK w podstawie programowej*. Dyrektor Szkoły, nr 3, s. 42-45.

Rostkowska, M. (2019). *Model SAMR a TIK: nauczyciel buduje kompetencje swoje i swoich uczniów*. Meritum, nr 1, s. 95-101.

Syśło, M. (2014). *Myślenie komputacyjne. Nowe spojrzenie na kompetencje informatyczne*. Materiały konferencyjne, „Informatyka w Edukacji, XI”, UMK Toruń, s. 15-32.

Syśło, M.M. Syśło. (2014). *Dość technologii, zaczynamy wszystkich uczniów uczyć informatyki i programowania*. Dostępny w:

http://www.kassk.pl/NTomysl_2014_MMSyslo.pdf [Dostęp: 06.04.2022].

Wing, J. M. (2008). *Computational thinking and thinking about computing*. Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences, 366(1881), 3717-3725.

Kucner, A., Pacewicz, G., Wasyluk, P., 2020. *Edukacja przyszłości 2020 - raport*. Olsztyn: Uniwersytet Warmińsko-Mazurski.

dr Jarosław Drapała, Instytut Informatyki, Politechnika Wrocławska. Dostępny w:

http://akademia.wroc.pl/pl/?page_id=137

<https://www.wroclaw.pl/jezyk-maszyn> [Dostęp: 08.03.2022].

Pitler, H., Hubbell, E.R., Kuhn, M., 2015. *Efektywne wykorzystanie nowych technologii na lekcjach*, przeł. Patrycja Szmyd, Warszawa: Centrum Edukacji Obywatelskiej, sygn. 32944-945.

Borawska-Kalbarczyk, K., (2015). *Kompetencje informacyjne uczniów w perspektywie zmian szkolnego środowiska uczenia*, Warszawa: Wydawnictwo Akademickie Żak, sygn. 32974.



Ostrowska, M., Sterna, D., (2015). *Technologie informacyjno-komunikacyjne na lekcjach. Przykładowe konspekty i polecane praktyki*, Warszawa: Centrum Edukacji Obywatelskiej.

Zbiór rozpraw pod redakcją Kozubska, A., Maciołek R., Ziółkowski P., (2017). *ZESZYTY NAUKOWE WSG, t. 29 seria: Edukacja – Rodzina – Społeczeństwo*.

Monk S., (2018). *Arduino dla początkujących. Podstawy i szkice*, Helion, Gliwice.

Rada ds. Informatyzacji Edukacji przy Ministrze Edukacji Narodowej, Podstawa programowa kształcenia informatycznego. Propozycja zmian w obowiązującej podstawie programowej. Dostępny w: <https://men.gov.pl/wp-content/uploads/2015/07/propozycja-zmian-w-podstawie-programowej.pdf> [Dostęp: 04.04.2022].

<https://programowanie.men.gov.pl/powstala-cyfrowa-mapa-szkol-z-dostepem-do-internetu/> [Dostęp: 08.03.2022]

<https://www.tiobe.com/tiobe-index/> [Dostęp: 03.03.2022]

BBC micro:bit, „Linux Magazine” 2017, nr 11 (165).

<http://www.arduino.cc/en/Main/Education>.



WYKAZ TABEL

Tabela 1. Kapitał ludzki	12
--------------------------------	----

WYKAZ ILUSTRACJI

Rysunek 1. Sekwencja świateł drogowych	23
Rysunek 2. Przykład sygnalizacji świetlnej z sekundnikiem	25
Rysunek 3. Schemat sterowania sygnalizacją świetlną	25
Rysunek 4. Sposób podłączenia diod led do płytki micro:bit	27
Rysunek 5. Montaż elementów na płytce stykowej	27
Rysunek 6. Program symulatora świateł drogowych w języku blokowym	28
Rysunek 7. Program symulatora świateł drogowych z sekundnikiem w języku blokowym	29
Rysunek 8. Schemat podłączenia układu do Arduino	30
Rysunek 9. Schemat elektryczny podłączenia diod led do Arduino	31
Rysunek 10. Schemat podłączenia modułu sygnalizatora świetlnego do Arduino	34